Neural Image Caption Generator

Project Report

Under the guidance of Dr.Deepak Singh Tomar(Associate Professor,CSE,M.A.N.I.T,Bhopal)

Submitted By:
Akashdeep Goel
(B.Tech CSE, I.I.I.T Gwalior)

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

I am highly indebted to Dr. Deepak Tomar sir for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

I would like to express our gratitude towards our parents for their kind cooperation and encouragement which helped us in completion of this project.

My thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

INDEX

ABSTRACT

There are many method available which can produce image captions,but there also remains a confusion when there are a lot of choices available as there remains a confusion that which model can best analyze the image and can intelligently generate caption for it.

In this project multiple models have been built , which differentiate on the CNN layers used among them.The concepts of deep neural network, recurrent neural network, convolutional neural network and transfer learning have been used.

For training and testing purposes commonly available Flickr dataset is used which has resemblance to common things will make the model more generalised and hence the study. The dataset is available at Kaggle
https://www.kaggle.com/shadabhussain/flickr8k

There are many factors which can influence the caption generation and thereby making it quite complex. During the study I have tried to minimize the effect of external factors by providing the same environment in terms of image dimension, number of epochs etc so that the result remains unbiased.

Chapter 1

INTRODUCTION

1.1 Objective

- To develop multiple deep learning models using same recurrent neural network and different types of CNN layers to automatically generate captions for the image.
- To compare the accuracy of the outputs generated by different models, one of which uses ResNet 50 and other uses Inception V3.

1.2 Background

- The accuracy of generating captions depends on human expertise in order to generate or verify them.
- A wide variety of companies like microsoft‘s captionbot ai uses a deep learning approach to automatically generate captions.
- Image captioning is important for automatic image indexing, which is important for Content-Based Image Retrieval (CBIR) .

1.3 Motivation

- The recent boom in the field of deep learning especially CNN has helped in creating better feature extraction models.
- In recent years there are various state of the art pretrained models developed for handling images like resnet, vg, mobilenet etc.

1.4 Python

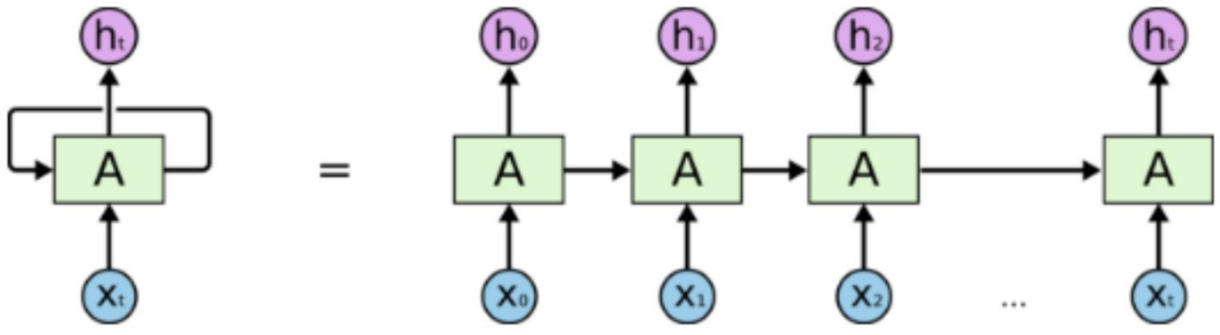
Python language is used to implement the project. Python is an “interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together”.

Python's simple programming language, easy to learn syntax have readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

Chapter 2

Used Algorithm and Methodology

2.1 Recurrent Neural Network



An unrolled recurrent neural network.

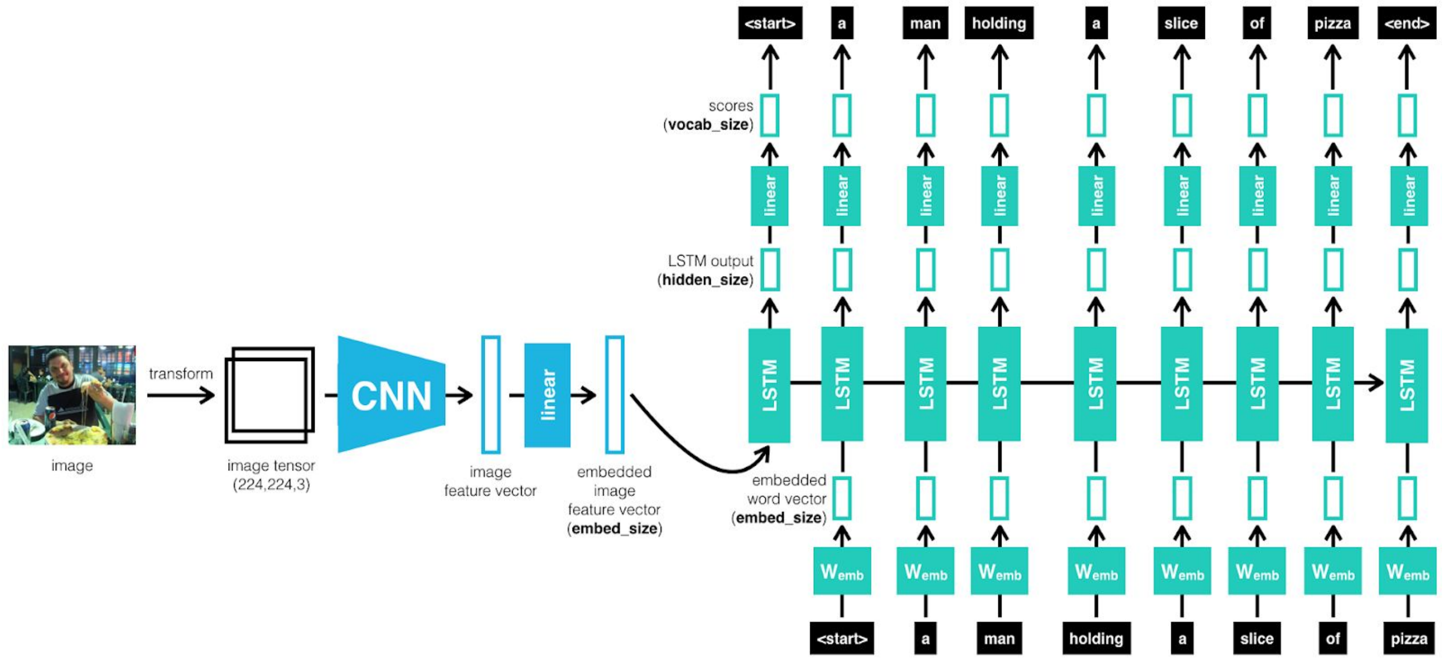
A RNN model referred as Recurrent neural network is used in the calculation of activation vectors(i.e feature map). The input is first fed to the embedding layer from that we get a metric of activation then this metric fed to the first block of Recurrent neural network and then it outputs a word with maximum likelihood.

Then this word with maximum likelihood is multiplied to the activation matric of the previous state and this gives the activation matric of the current state.

This function is repeated till we get an end sequence and maximum likelihood estimated value.

Recurrent neural networks a particularly long short-term memory to give an example of how long short-term memory works we will consider the question of what's for dinner let's say for a minute that you are a very lucky apartment dweller and you have a flatmate who loves to cook dinner every night he cooks one of three things

For the representation of images, we use a CNN. CNNs are primarily used in the study of images, and are currently the best for object identification and detection. Our particular choice of CNN uses a approach of batch normalization and produces the current best performance in the ILSVRC 2014 classification competition



2.2 Convolutional Neural Network

A CNN is a deep neural network that is used for image extraction tasks in deep learning. There are multiple types of layers used in CNNs, some of which are listed below-

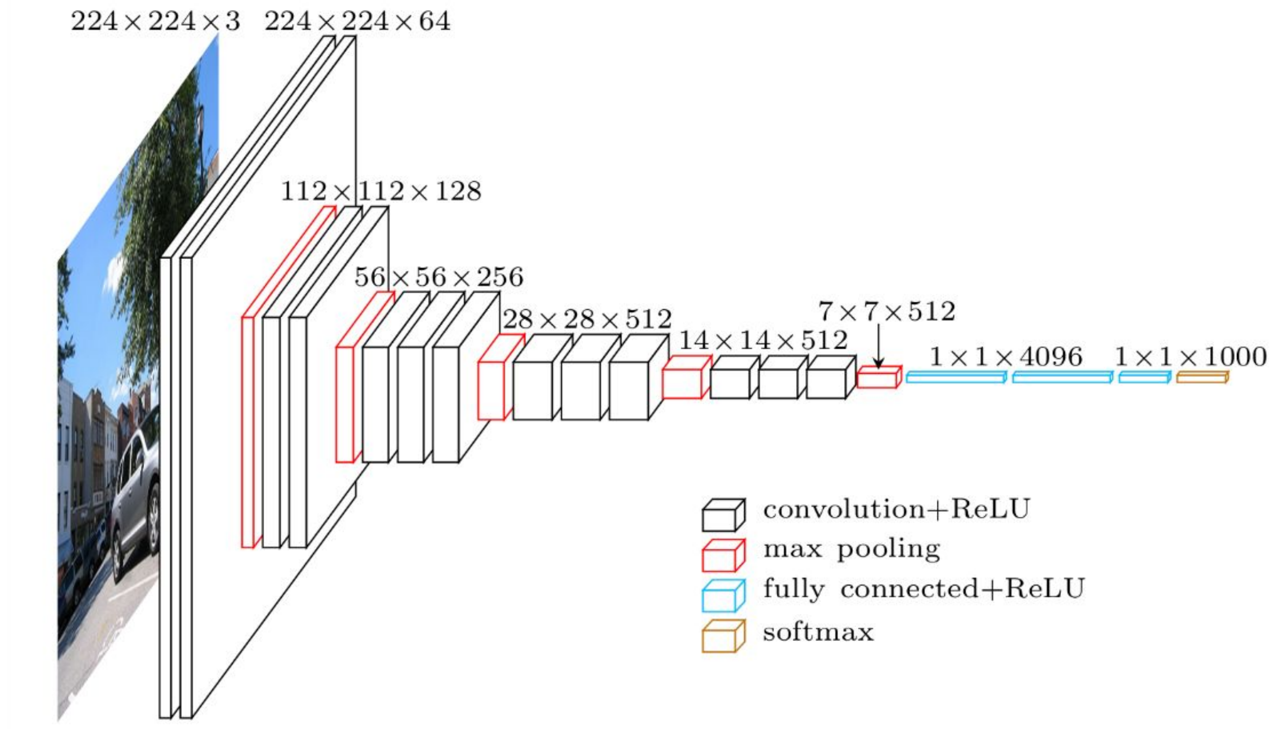
- **Input Layer:** " It is made up of artificial input neurons, and brings the initial data into the system for processing by the following layers of artificial neurons. The input layer is the starting of the workflow in the neural network."
- **Hidden Layers: "**It is situated between input and output layers of a neural network, hidden layers perform non linear transformations of the inputs coming in the network and the function applies weights to the inputs and directs them through an activation function as output.**"**
- **Output Layer: "** In an artificial neural network, the output layer is the final layer of neurons that produces required outputs for the neural network in form of probabilities."
- **Activation Function: "**It tells us that should a neuron be activated or not by finding

a weighted sum and further adding bias to it."

- **Batch normalization: "**It is a technique used for training large neural networks that treats the inputs as a mini-batch. This has the effect of stabilizing the learning process and reducing the number of training epochs required to train large networks."
- **Average pooling: "**it performs down-sampling by breaking down the input into pooling regions and computing the average values over each pooling region."
- **Convolution: "**It is performed on the input data with the use of a filter or kernel by sliding the filter over the input to then produce a feature map."
- **Fully Connected - "**It is a typical layer which is obtained from previous layer by using a parameter matrix."

2.3 Image feature Extraction using Transfer learning

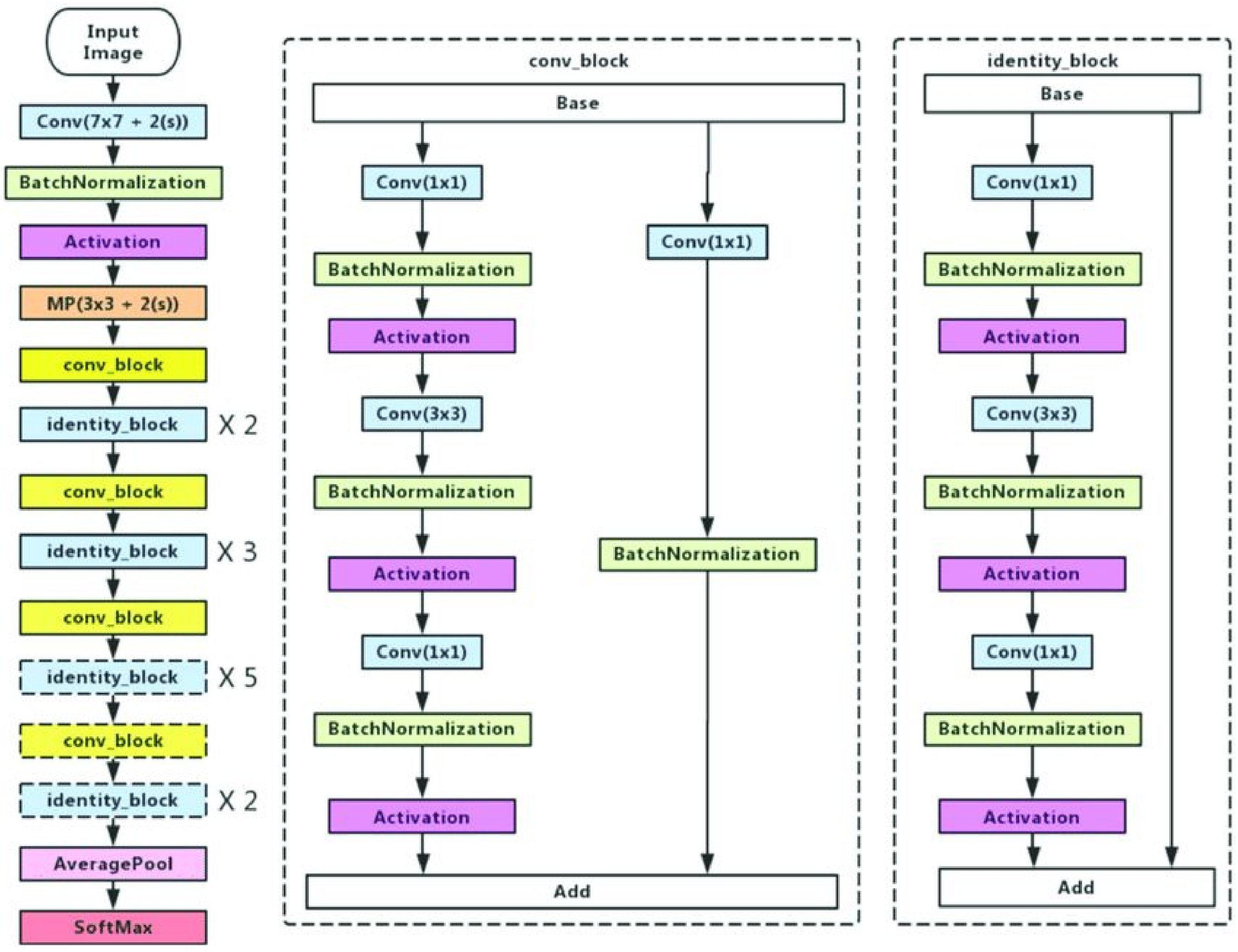
Transfer learning a technique in which we extract the features from the target with the help of a pretrained model. The pretrained means the model is previously trained on the particular dataset and hyperparameters and weights are stored. So that these models can be reused for feature extraction.Like in this we are using a Pretrained Resnet model which was trained on imagenet dataset.



2.4 Resnet model

Following are the characteristics of the Resnet architecture

- Input size : 224x224x3
- Activation vector size: 7x7x512
- res5c_branch2c (Conv2D) : (None, 7, 7, 2048)
- output Gap layer : (None, 2048)

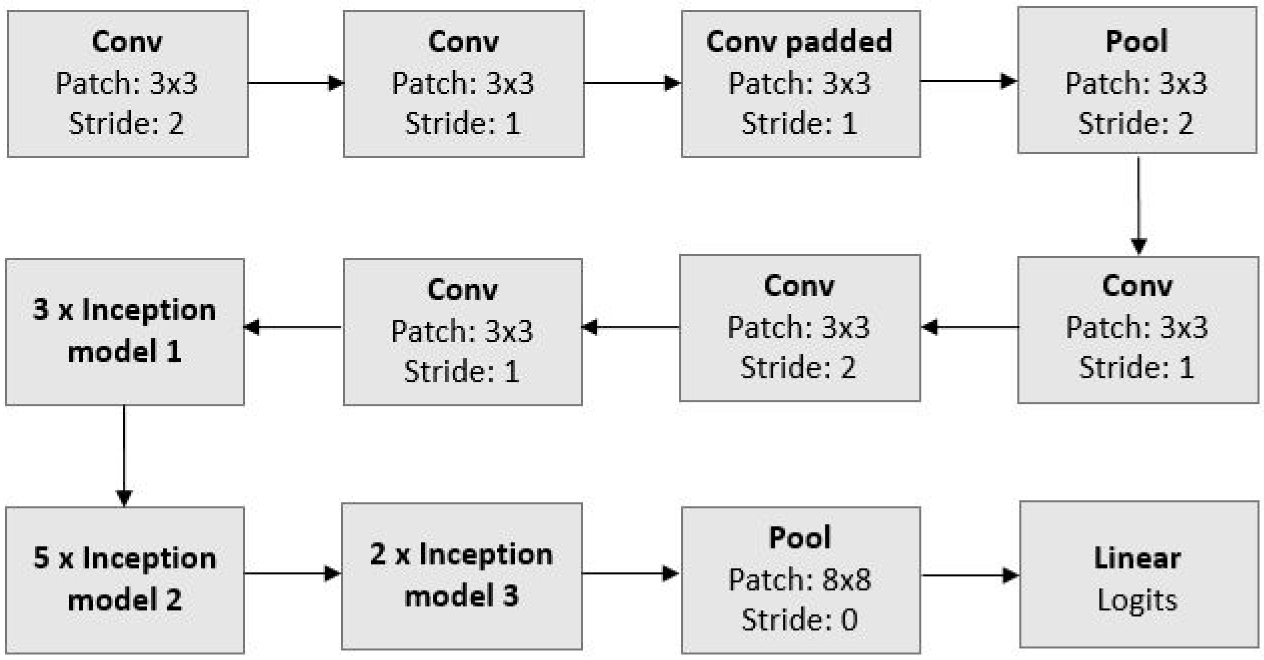


Architecture of Resnet 50

2.5 Inception V3 model

Following are the prime characteristics of Inception V3 architecture:

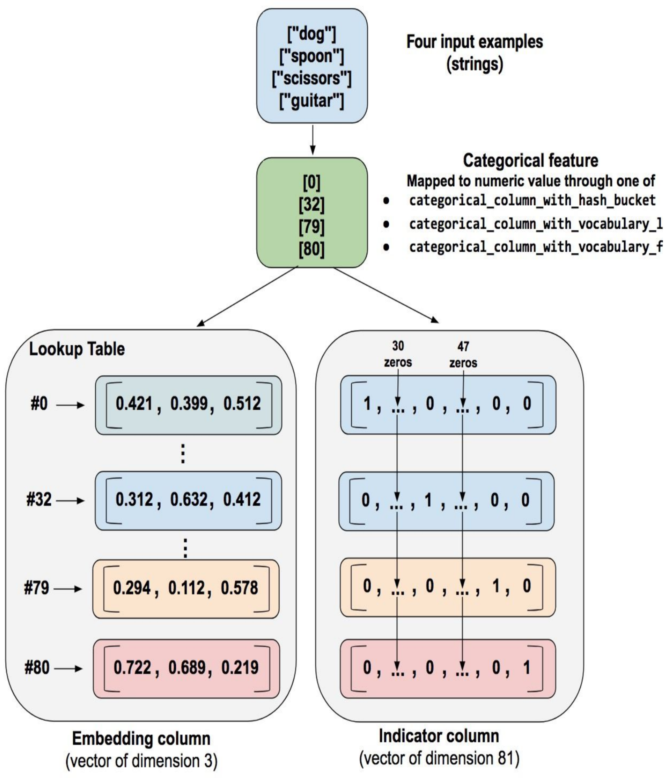
- RMSProp Optimizer
- Factorized 7x7 convolutions.
- BatchNorm in the Auxiliary Classifiers
- Label Smoothing (A type of regularizing component added to the loss formula that prevents the network from becoming too confident about a class. Prevents over fitting).



2.6 Text Feature Engineering using Glove vector

Word Embeddings are used to "convert texts into numerical representation".These numerical values are close for a given category/set.

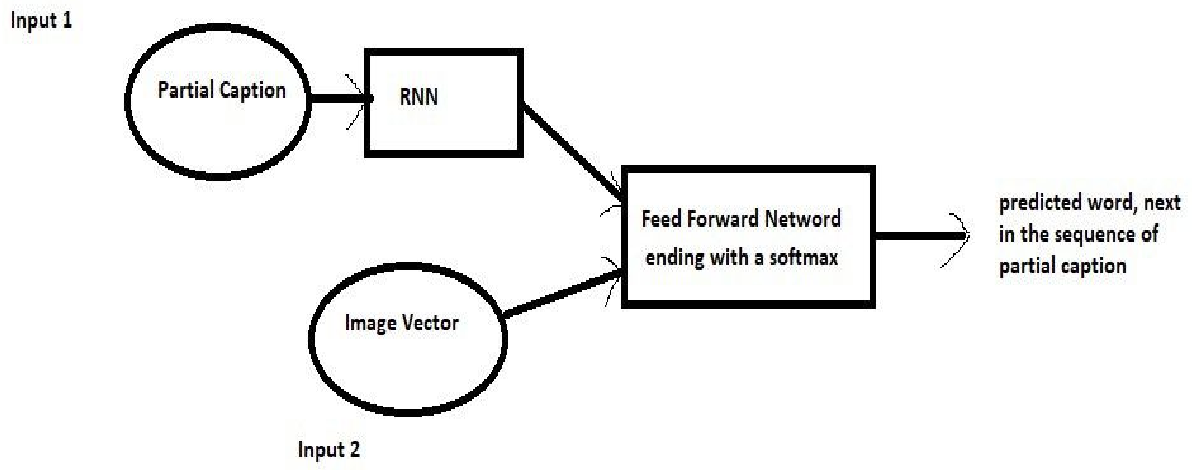
.



For a given image

- Train : 5 sentences
- Total Words 373837
- Total unique word (vocab):8424
- Reducing vocab size
- Thresholding : 1652+2
- Embedding matrix shape: (1654, 200)
- Max length of sentence : 34

2.7 Model Architecture


Input 1
Partial Caption
RNN
Feed Forward Netword ending with a softmax
predicted word, next in the sequence of partial caption
Image Vector
Input 2

General architecture of the RNN model Implemented

Chapter 3

Implementation and Outputs

3.1 Used Python Libraries

- **Numpy**: Numpy is a python package used for scientific computing.It is used to process multidimensional arrays, and also used for various numerical operations.

- **Pandas:** Pandas is a powerful python library which has Data frames its main data structure. It is used to manipulate data into rows and columns.

- **Matplotlib:** It is a python library used for visualization. It is used for plotting graphs, charts etc.

- **Scikit-learn:** It is a free python library. It consists of vast num-ber of methods. It supports many machine learning algorithms like SVM, KNN, Decision Tree, K-Means Clustering, etc. It also supports numerical computations like calculating errors, accuracy, matrix, etc.

- **OpenCV:** It is the Open Source Computer Vision library. It is a li-brary containing programming functions that support real-time com-puter vision and image processing. It is written in C++. It is developed by Intel later supported by Willow Garage.

- **Keras:** It is basically a high level API of Tensorflow that is used for hifh level and state of the art research. It is efficient fast and user friendly.

- **Tensorflow:** It is an open-source library primarily used for high level mathematics and machine learning. It was developed by Google.

3.2 Implementation Steps

1. Reading captions
2. Converting captions to dictionary with image ID as key.
3. Text cleaning(removing punctuations,numbers etc) but keeping stemming words
4. Creating a list of unique words named as vocab
5. Removing words of frequency less than a threshold
6. Preparing dictionary named train_desc for mapping image ID to caption for training set.
7. Appending startseq and endseq keywords.
8. Preprocessing image using inbuilt function and extracting feature using Resnet50/Inception V3
9. Preparing dictionary encoding train and encoding test to map features and image ID of an image.
10. Like image was converted to numbers in previous step , we convert word to numbers(indexes)
11. As stated in previous step embedding matrix of vocab is created using glove 50
12. Now feeding the above developed data structures to the neural network model architecture shown under the category model architecture.

The dataset used for the purpose of image captioning is available at Kaggle:
https://www.kaggle.com/shadabhussain/flickr8k

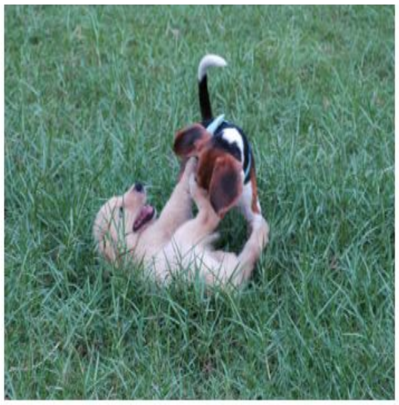
3.3 Source Code

Although The Complete Code is Very Long and Can't be Completely Written in This Document ,Here are Main Model architecture

```
“”input_img_features = Input(shape=(2048,))
inp_img1 = Dropout(0.3)(input_img_features)
inp_img2 = Dense(256,activation='relu')(inp_img1)

# Captions as Input
input_captions = Input(shape=(max_len,))
inp_cap1 = Embedding(input_dim=vocab_size,output_dim=50,mask_zero=True)(input_captions)
inp_cap2 = Dropout(0.3)(inp_cap1)
inp_cap3 = Long Short Term Memory networks(256)(inp_cap2)
decoder1 = add([inp_img2,inp_cap3])
```

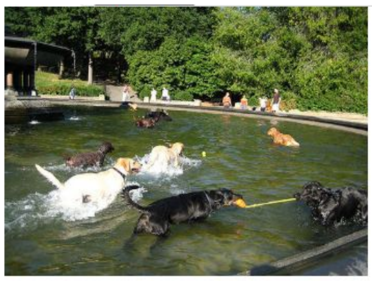
```
decoder2 = Dense(256,activation='relu')(decoder1)
outputs = Dense(vocab_size,activation='softmax')(decoder2)

# Combined Model
model = Model(inputs=[input_img_features,input_captions],outputs=outputs)
model.compile(loss='categorical_crossentropy',optimizer="adam")
epochs = 20
batch_size = 3
number_pics_per_batch = 60
steps = len(train_descriptions)//number_pics_per_batch
def train():
    for i in range(epochs):
        generator = data_generator(train_descriptions,encoding_train,word_to_idx,max_len,batch_size)
        model.fit_generator(generator,epochs=1,steps_per_epoch=steps,verbose=1)””
```

3.4 Outputs



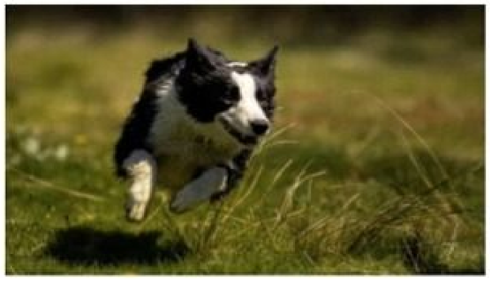
Generated caption :

white dog is running on the grass



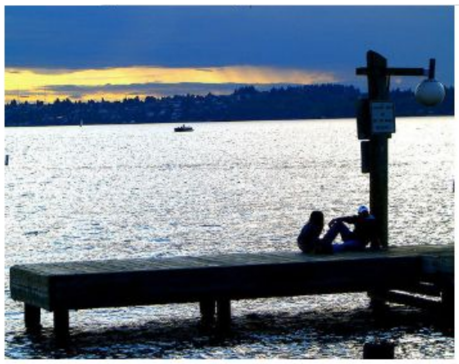
Generated caption :

two dogs are playing in the water



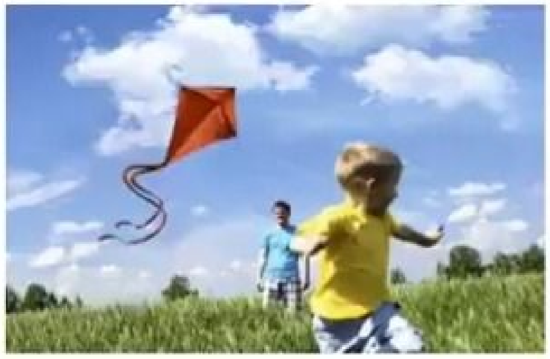
Generated Caption :

dog is running through the grass



Generated caption :

two people sitting on dock near the sunset



Generated Caption :

boy in red shirt is running through the grass

Chapter 4
RESULT AND EVALUATION

4.1 Categorical Cross Entropy Loss Function

"It is the difference between the true value and predicted value. A loss function is for a single training example. It is also sometimes called an error function. It is a Softmax activation plus a Cross-Entropy loss. If we use this loss, we will train a CNN to output a probability over the classes for each image."

The main advantage of categorical loss function over other loss function like root mean square etc is that it is a non convex function, i.e exactly one minima.

Thus the gradient descent when applied is does not settle for local minimum loss and finds global minimum loss. Also, it solves the initialisation problem of weights.

Mathematically the loss(J) is given as:

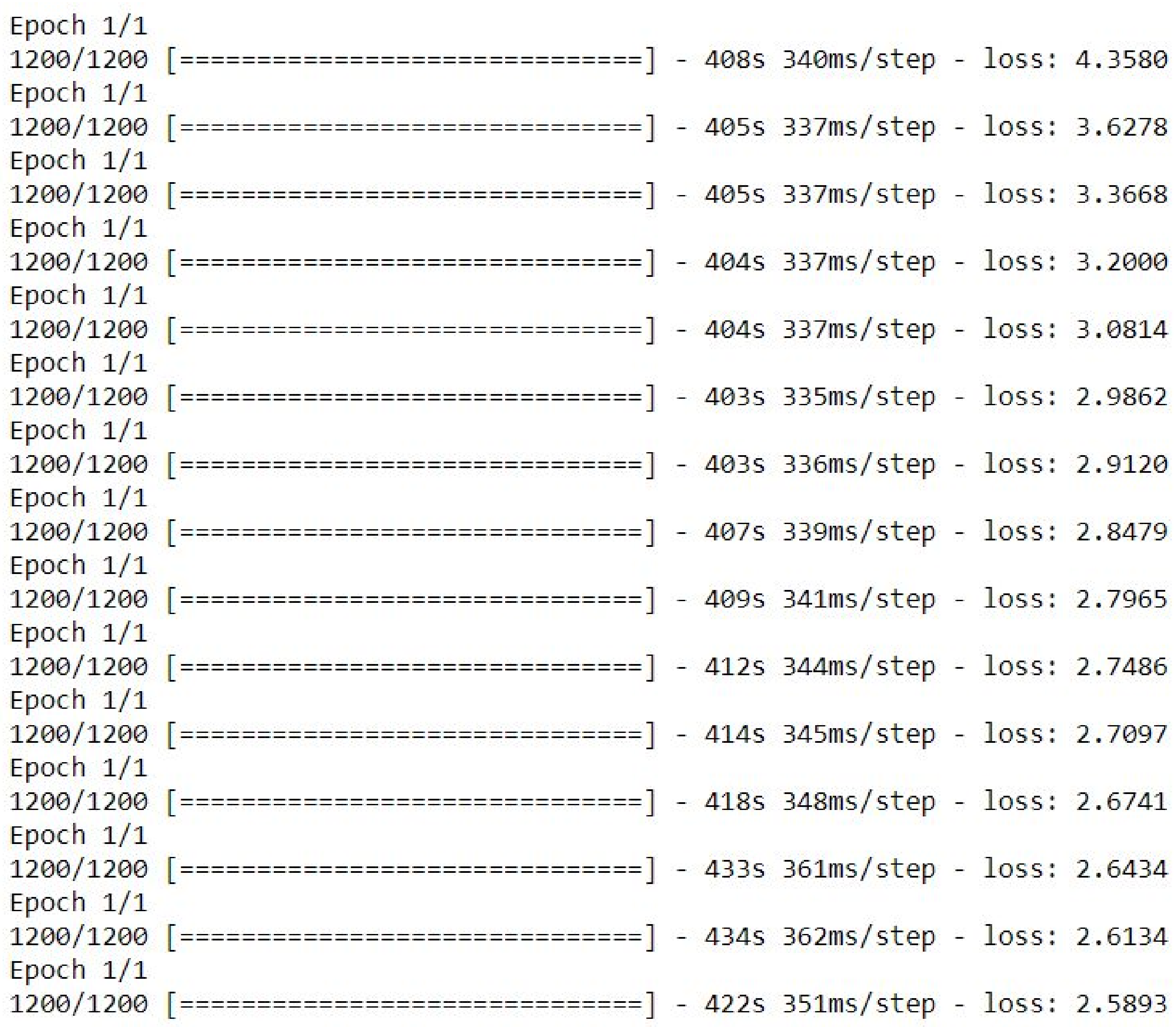
$$J = -\frac{1}{N}\left(\sum_{i=1}^{N} \mathbf{y_i} \cdot \log(\hat{\mathbf{y}}_\mathbf{i})\right)$$

where,

N= Number of output classes

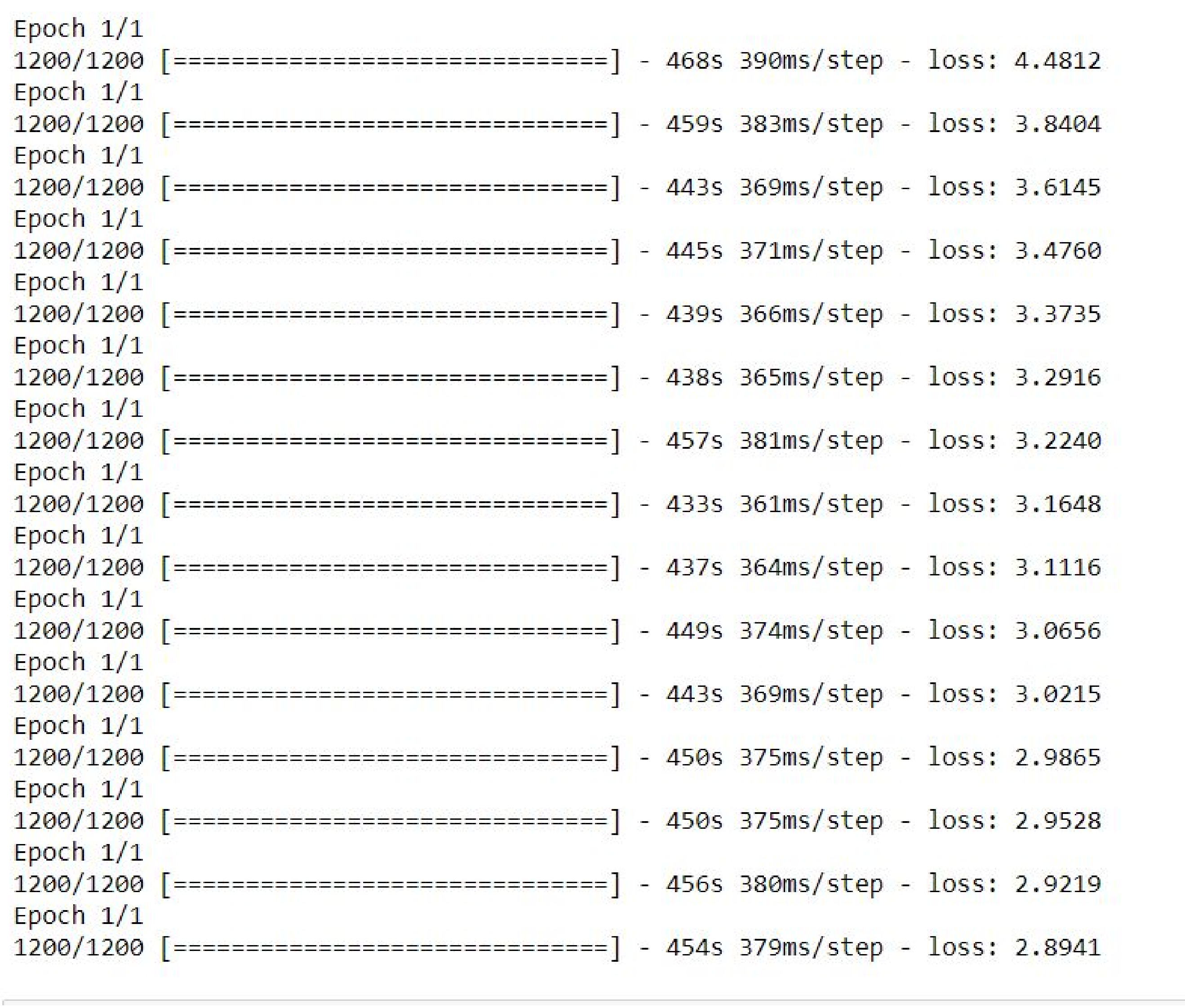
y= Actual output

y^= Generated output



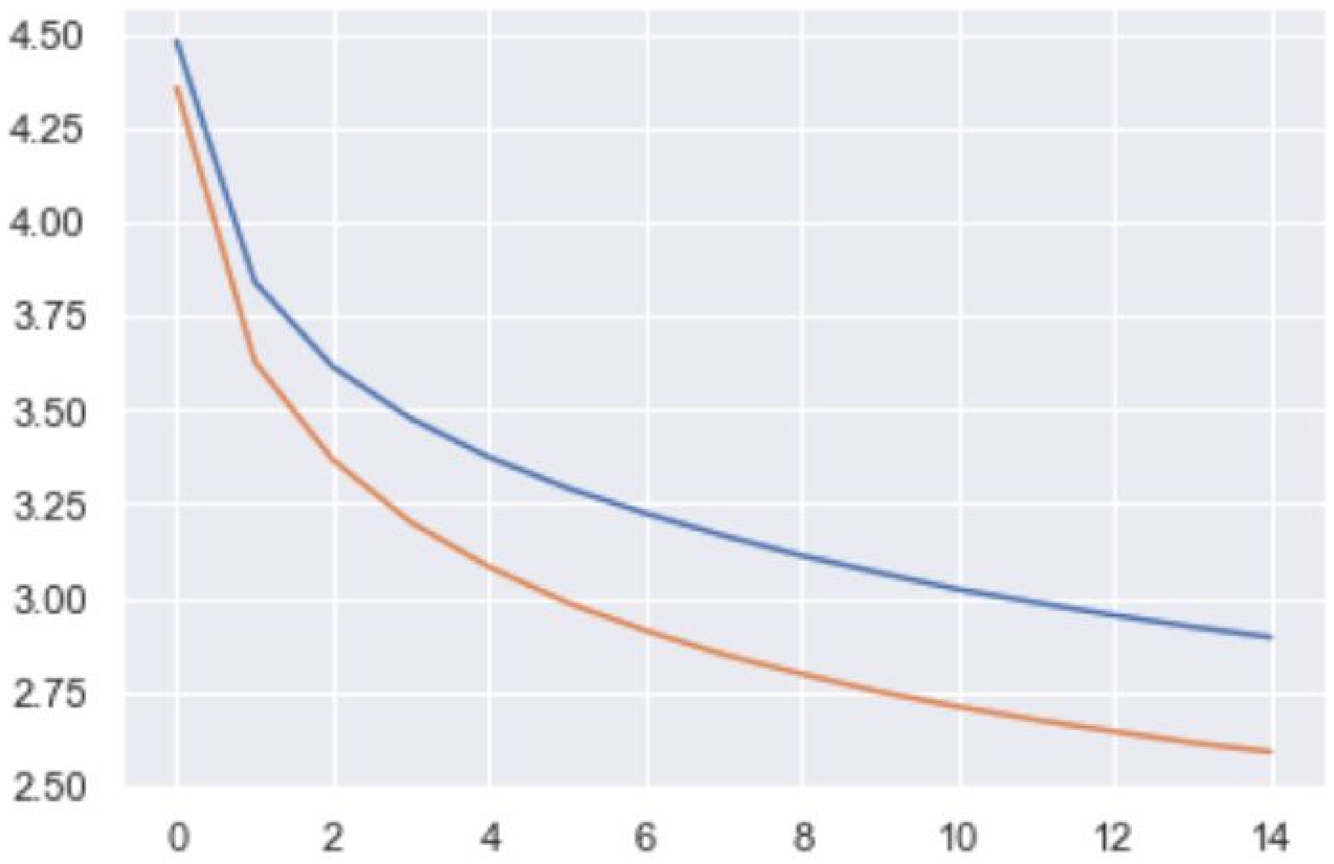
```
Epoch 1/1
1200/1200 [==============================] - 408s 340ms/step - loss: 4.3580
Epoch 1/1
1200/1200 [==============================] - 405s 337ms/step - loss: 3.6278
Epoch 1/1
1200/1200 [==============================] - 405s 337ms/step - loss: 3.3668
Epoch 1/1
1200/1200 [==============================] - 404s 337ms/step - loss: 3.2000
Epoch 1/1
1200/1200 [==============================] - 404s 337ms/step - loss: 3.0814
Epoch 1/1
1200/1200 [==============================] - 403s 335ms/step - loss: 2.9862
Epoch 1/1
1200/1200 [==============================] - 403s 336ms/step - loss: 2.9120
Epoch 1/1
1200/1200 [==============================] - 407s 339ms/step - loss: 2.8479
Epoch 1/1
1200/1200 [==============================] - 409s 341ms/step - loss: 2.7965
Epoch 1/1
1200/1200 [==============================] - 412s 344ms/step - loss: 2.7486
Epoch 1/1
1200/1200 [==============================] - 414s 345ms/step - loss: 2.7097
Epoch 1/1
1200/1200 [==============================] - 418s 348ms/step - loss: 2.6741
Epoch 1/1
1200/1200 [==============================] - 433s 361ms/step - loss: 2.6434
Epoch 1/1
1200/1200 [==============================] - 434s 362ms/step - loss: 2.6134
Epoch 1/1
1200/1200 [==============================] - 422s 351ms/step - loss: 2.5893
```

Model trained for 15 epoch on Inception model



```
Epoch 1/1
1200/1200 [==============================] - 468s 390ms/step - loss: 4.4812
Epoch 1/1
1200/1200 [==============================] - 459s 383ms/step - loss: 3.8404
Epoch 1/1
1200/1200 [==============================] - 443s 369ms/step - loss: 3.6145
Epoch 1/1
1200/1200 [==============================] - 445s 371ms/step - loss: 3.4760
Epoch 1/1
1200/1200 [==============================] - 439s 366ms/step - loss: 3.3735
Epoch 1/1
1200/1200 [==============================] - 438s 365ms/step - loss: 3.2916
Epoch 1/1
1200/1200 [==============================] - 457s 381ms/step - loss: 3.2240
Epoch 1/1
1200/1200 [==============================] - 433s 361ms/step - loss: 3.1648
Epoch 1/1
1200/1200 [==============================] - 437s 364ms/step - loss: 3.1116
Epoch 1/1
1200/1200 [==============================] - 449s 374ms/step - loss: 3.0656
Epoch 1/1
1200/1200 [==============================] - 443s 369ms/step - loss: 3.0215
Epoch 1/1
1200/1200 [==============================] - 450s 375ms/step - loss: 2.9865
Epoch 1/1
1200/1200 [==============================] - 450s 375ms/step - loss: 2.9528
Epoch 1/1
1200/1200 [==============================] - 456s 380ms/step - loss: 2.9219
Epoch 1/1
1200/1200 [==============================] - 454s 379ms/step - loss: 2.8941
```

Model trained for 15 epoch on Resnet50 model



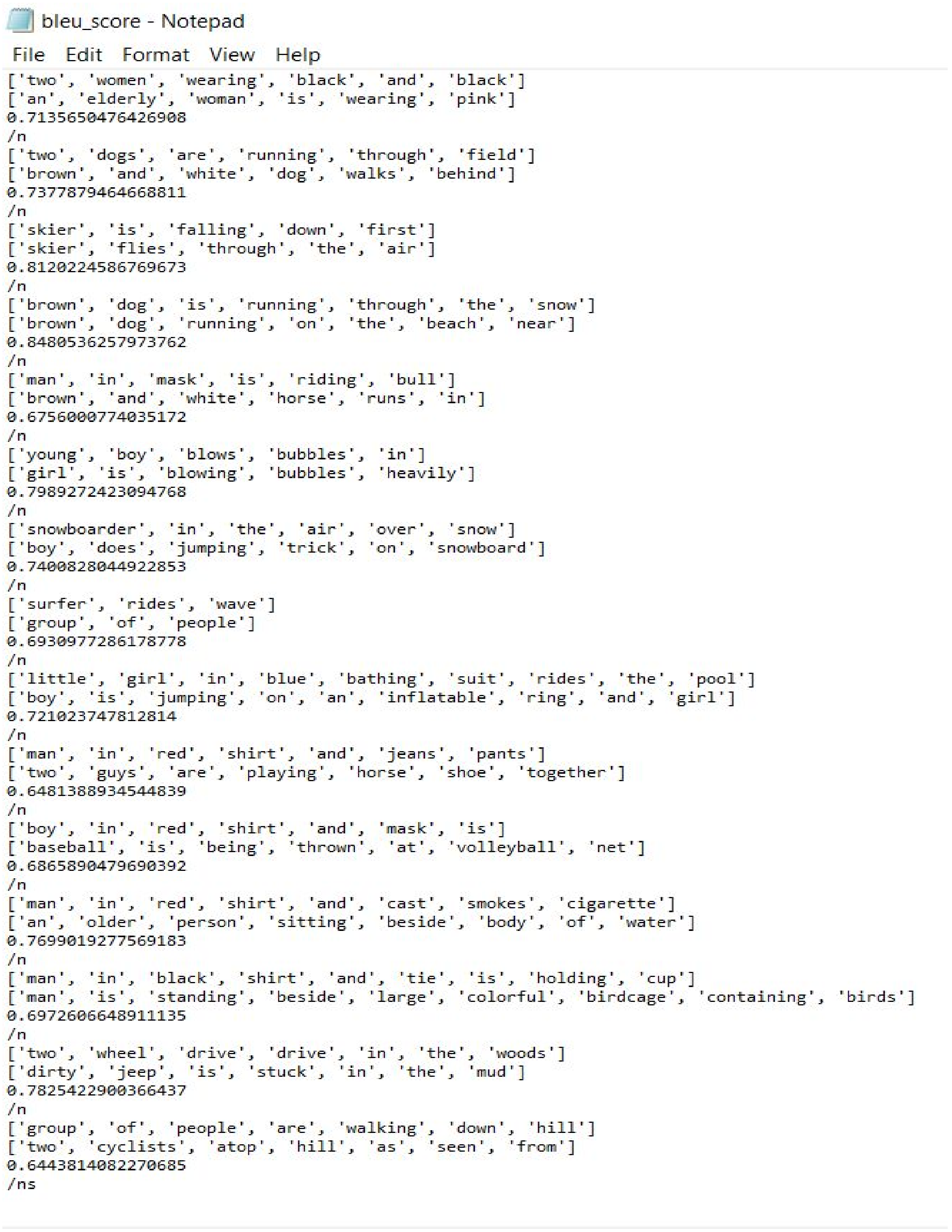
Comparison for 15 epoch of Loss function between Inception and Resnet50

The Blue line indicates loss for Resnet

The Red line indicates loss for Inception

4.2 Evaluation Metric

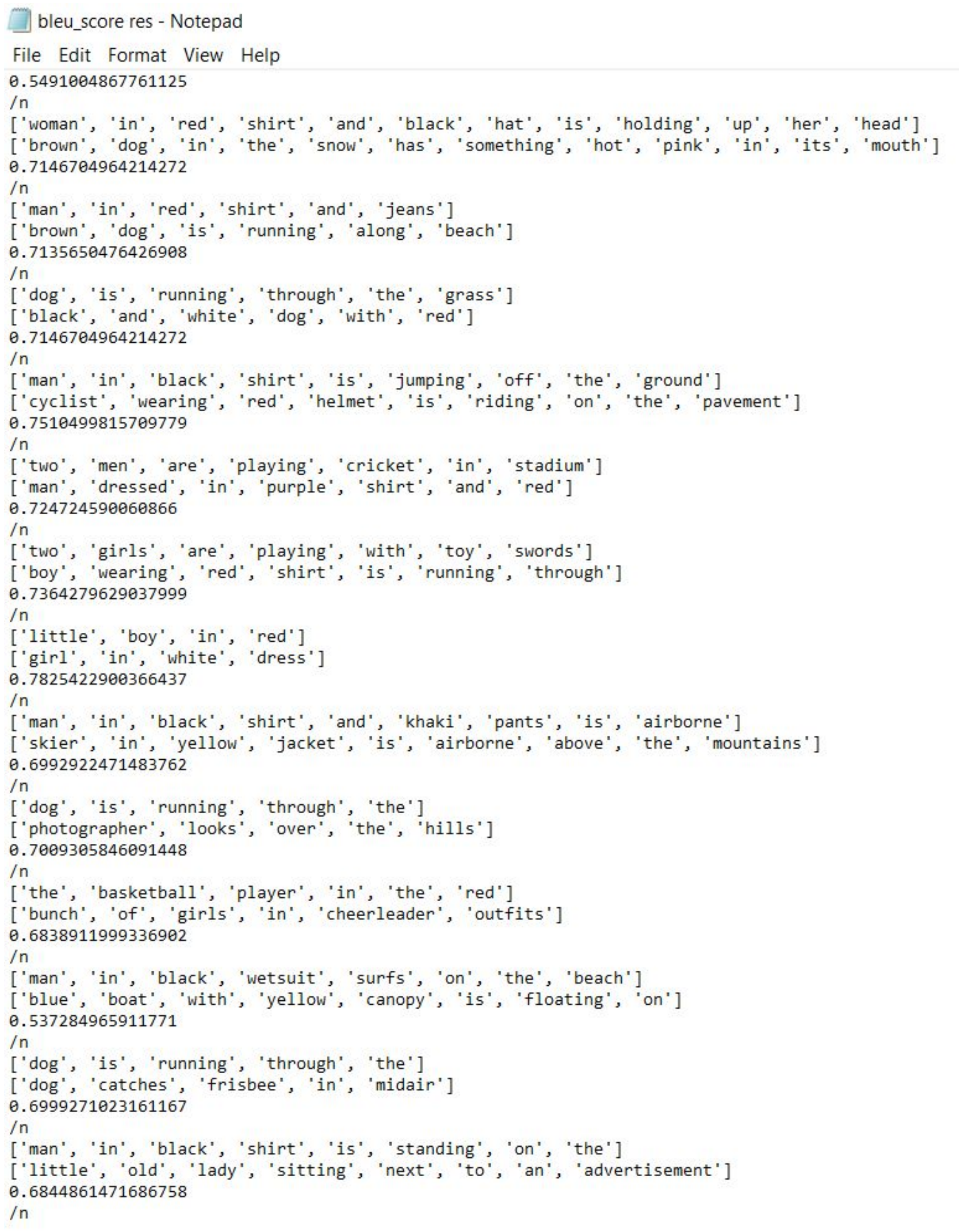
4.2.1 BLEU SCORE: It stands for "Bilingual Evaluation Understudy **Score". "The BLEU score was proposed by Kishore Papineni, et al. in their 2002 paper". "BLEU: a Method for Automatic Evaluation of Machine Translation**". It is a metric for evaluating a generated sentence to a reference sentence. For a perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0.0."


bleu_score - Notepad

File Edit Format View Help

```
['two', 'women', 'wearing', 'black', 'and', 'black']
['an', 'elderly', 'woman', 'is', 'wearing', 'pink']
0.7135650476426908
/n
['two', 'dogs', 'are', 'running', 'through', 'field']
['brown', 'and', 'white', 'dog', 'walks', 'behind']
0.7377879464668811
/n
['skier', 'is', 'falling', 'down', 'first']
['skier', 'flies', 'through', 'the', 'air']
0.8120224586769673
/n
['brown', 'dog', 'is', 'running', 'through', 'the', 'snow']
['brown', 'dog', 'running', 'on', 'the', 'beach', 'near']
0.8480536257973762
/n
['man', 'in', 'mask', 'is', 'riding', 'bull']
['brown', 'and', 'white', 'horse', 'runs', 'in']
0.6756000774035172
/n
['young', 'boy', 'blows', 'bubbles', 'in']
['girl', 'is', 'blowing', 'bubbles', 'heavily']
0.7989272423094768
/n
['snowboarder', 'in', 'the', 'air', 'over', 'snow']
['boy', 'does', 'jumping', 'trick', 'on', 'snowboard']
0.7400828044922853
/n
['surfer', 'rides', 'wave']
['group', 'of', 'people']
0.6930977286178778
/n
['little', 'girl', 'in', 'blue', 'bathing', 'suit', 'rides', 'the', 'pool']
['boy', 'is', 'jumping', 'on', 'an', 'inflatable', 'ring', 'and', 'girl']
0.721023747812814
/n
['man', 'in', 'red', 'shirt', 'and', 'jeans', 'pants']
['two', 'guys', 'are', 'playing', 'horse', 'shoe', 'together']
0.6481388934544839
/n
['boy', 'in', 'red', 'shirt', 'and', 'mask', 'is']
['baseball', 'is', 'being', 'thrown', 'at', 'volleyball', 'net']
0.6865890479690392
/n
['man', 'in', 'red', 'shirt', 'and', 'cast', 'smokes', 'cigarette']
['an', 'older', 'person', 'sitting', 'beside', 'body', 'of', 'water']
0.7699019277569183
/n
['man', 'in', 'black', 'shirt', 'and', 'tie', 'is', 'holding', 'cup']
['man', 'is', 'standing', 'beside', 'large', 'colorful', 'birdcage', 'containing', 'birds']
0.6972606648911135
/n
['two', 'wheel', 'drive', 'drive', 'in', 'the', 'woods']
['dirty', 'jeep', 'is', 'stuck', 'in', 'the', 'mud']
0.7825422900366437
/n
['group', 'of', 'people', 'are', 'walking', 'down', 'hill']
['two', 'cyclists', 'atop', 'hill', 'as', 'seen', 'from']
0.6443814082270685
/ns
```

BLEU-score for Inception model



```
0.5491004867761125
/n
['woman', 'in', 'red', 'shirt', 'and', 'black', 'hat', 'is', 'holding', 'up', 'her', 'head']
['brown', 'dog', 'in', 'the', 'snow', 'has', 'something', 'hot', 'pink', 'in', 'its', 'mouth']
0.7146704964214272
/n
['man', 'in', 'red', 'shirt', 'and', 'jeans']
['brown', 'dog', 'is', 'running', 'along', 'beach']
0.7135650476426908
/n
['dog', 'is', 'running', 'through', 'the', 'grass']
['black', 'and', 'white', 'dog', 'with', 'red']
0.7146704964214272
/n
['man', 'in', 'black', 'shirt', 'is', 'jumping', 'off', 'the', 'ground']
['cyclist', 'wearing', 'red', 'helmet', 'is', 'riding', 'on', 'the', 'pavement']
0.7510499815709779
/n
['two', 'men', 'are', 'playing', 'cricket', 'in', 'stadium']
['man', 'dressed', 'in', 'purple', 'shirt', 'and', 'red']
0.724724590060866
/n
['two', 'girls', 'are', 'playing', 'with', 'toy', 'swords']
['boy', 'wearing', 'red', 'shirt', 'is', 'running', 'through']
0.7364279629037999
/n
['little', 'boy', 'in', 'red']
['girl', 'in', 'white', 'dress']
0.7825422900366437
/n
['man', 'in', 'black', 'shirt', 'and', 'khaki', 'pants', 'is', 'airborne']
['skier', 'in', 'yellow', 'jacket', 'is', 'airborne', 'above', 'the', 'mountains']
0.6992922471483762
/n
['dog', 'is', 'running', 'through', 'the']
['photographer', 'looks', 'over', 'the', 'hills']
0.7009305846091448
/n
['the', 'basketball', 'player', 'in', 'the', 'red']
['bunch', 'of', 'girls', 'in', 'cheerleader', 'outfits']
0.6838911999336902
/n
['man', 'in', 'black', 'wetsuit', 'surfs', 'on', 'the', 'beach']
['blue', 'boat', 'with', 'yellow', 'canopy', 'is', 'floating', 'on']
0.537284965911771
/n
['dog', 'is', 'running', 'through', 'the']
['dog', 'catches', 'frisbee', 'in', 'midair']
0.6999271023161167
/n
['man', 'in', 'black', 'shirt', 'is', 'standing', 'on', 'the']
['little', 'old', 'lady', 'sitting', 'next', 'to', 'an', 'advertisement']
0.6844861471686758
/n
```

BLEU-score for Resnet50 mode

Average bleu-score for the Inception model turns out to be <u>0.731262</u>.

Average bleu-score for the Resnet50 model turns out to be <u>0.69103</u>.

Chapter 5

CONCLUSION

1. Transfer learning algorithm is used along with image processing techniques.
2. The method has low computational cost and is suitable for detecting features with reasonable accuracy and is suitable for image preprocessing.
3. Supervised methods (including Recurrent neural networks) require quality training data to build a model that can perform well on testing data. Partial captions are used for training
4. The Bleu score for the Inception model is higher than Resnet 50 thus , we can conclude that Inception as a feature extractor for images is better than Resnet 50 for image captioning purposes.

FUTURE SCOPE

This Model can be useful in the following fields :-

1. **Help the Visually Impaired :** This technology can be embedded in a headband or glasses, which a visually impaired person can wear, a camera is connected to the headband or glasses. That camera can send the video stream of the surroundings environment to the model and then the model can predict the obstacles in the surroundings . This prediction can be converted to audio using TTS and can be heard by the person.
2. **Self Driving Cars :** All the companies who are building the self driving cars are using image processing and video processing with neural networks to attain their goal.
3. **Skin Cancer Prediction :** In the medical field this technology can be used in the prediction of skin cancer like classifying the tumor as Benign or Malignant.
4. **Classifying images in our Phone:** This technology can also be used for the classification of images in our Gallery for example classify in different categories like mountains, beach, portraits etc.

Chapter 6
References

1. M. Hodosh, P. Young, and J. Hockenmaier. *Framing image description as a ranking task: Data, models and evaluation metrics.* JAIR, 47, 2013.

2. S. Ioffe and C. Szegedy. Batch normalization: *Accelerating deep network training by reducing internal covariate shift*. In arXiv:1502.03167, 2015.

3. A. Karpathy, A. Joulin, and L. Fei-Fei. *Deep fragment embeddings for bidirectional image sentence mapping.* NIPS, 2014.

4.R. Kiros, R. Salakhutdinov, and R. S. Zemel. *Unifying visual-semantic embeddings with multimodal neural language models.* In arXiv:1411.2539, 2014.

5. R. Kiros and R. Z. R. Salakhutdinov. *Multimodal neural language models.* In NIPS Deep Learning Workshop, 2013. 2